

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-111960

(43)Date of publication of application : 20.04.2001

(51)Int.Cl.

H04N 5/92
G11B 20/12
G11B 20/18
G11B 27/00
G11B 27/10
H04N 5/765
H04N 5/781

(21)Application number : 11-288993

(71)Applicant : SHARP CORP

(22)Date of filing : 12.10.1999

(72)Inventor : KIYAMA JIRO
YAMAMURA HIROYUKI
YAMAGUCHI TAKAYOSHI

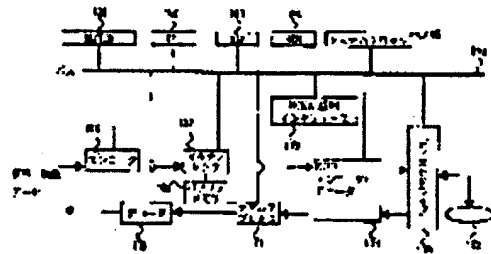
(54) DATA RECORDING METHOD AND DATA RESTORATION METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To solve a problem of conventional data recording and restoration methods that a time code recorded on a disk is in general expressed as a relative time from the head of video data by each video program and consequently a plurality of video data having the same time code may be in existence in the disk when video programs are recorded on the disk.

SOLUTION: In the case of recording video image or image data onto a recording medium, they are recorded on the recording medium being divided into a plurality of units, and information to identify a recording device is provided to each unit together with information denoting the reproduction sequence on each video sequence.

Through the above configuration, even when some units on the recording medium have the same information denoting the reproduction sequence on each video sequence in the case of reproduction, the device ID is provided uniquely to each device, and hence the video sequences can be identified.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2001-111960
(P2001-111960A)

(43) 公開日 平成13年4月20日 (2001.4.20)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード* (参考)
H 0 4 N 5/92		C 1 1 B 20/12	1 0 3 5 C 0 5 3
G 1 1 B 20/12	1 0 3	20/18	5 D 0 4 4
20/18		27/00	5 D 0 7 7
27/00		27/10	5 D 1 1 0
27/10		H 0 4 N 5/92	H
審査請求 未請求 請求項の数 6 O L (全 22 頁) 最終頁に続く			

(21) 出願番号 特願平11-288993

(22) 出願日 平成11年10月12日 (1999.10.12)

(71) 出願人 000005049

シャープ株式会社

大阪府大阪市阿倍野区長池町22番22号

(72) 発明者 木山 次郎

大阪府大阪市阿倍野区長池町22番22号 シ

ャープ株式会社内

(72) 発明者 山村 博幸

大阪府大阪市阿倍野区長池町22番22号 シ

ャープ株式会社内

(74) 代理人 100103296

弁理士 小池 隆彌

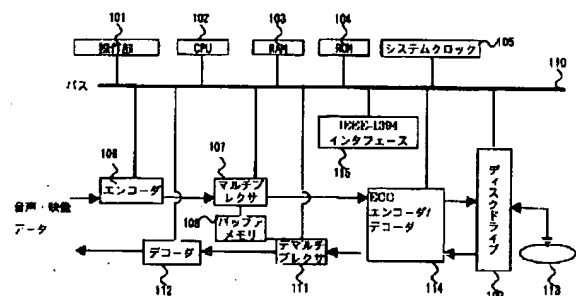
最終頁に続く

(54) 【発明の名称】 データ記録方法及びデータ復旧方法

(57) 【要約】

【課題】 一般的に、ディスク上に記録されるタイムコードは各ビデオプログラム毎のビデオデータの先頭からの相対時間で表される。したがって、ディスク中に複数のビデオプログラムを記録した場合、同じタイムコードを持つビデオデータが複数存在することになる。その場合、正しい順序で復元されないおそれがある。

【解決手段】 映像や画像データを記録媒体に記録する際に、記録媒体上では複数のユニットとして分割して記録し、このユニット内に、シーケンス上での再生順序を示す情報とともに、記録した機器を識別するための情報を付与する。この構成によって、再生時に、記録媒体上で、シーケンス上での再生順序を示す情報が同一のユニットが存在した場合においても、各機器にユニークに付与されている機器IDを付与しているため、シーケンスを区別することが可能である。



【特許請求の範囲】

【請求項1】 映像または音声データからなるシーケンスを、複数のユニットに分割して記録媒体に記録し、記録媒体の位置情報に対応させた前記ユニットの再生順を示す再生順管理データにより、記録媒体上に記録されたデータを管理する記録媒体におけるデータ記録方法であって、

各ユニットを再生すべきシーケンス上での順序を示す情報とともに、各ユニットを記録した機器を識別するための情報を各ユニットに記録することを特徴とするデータ記録方法。

【請求項2】 映像データまたは音声データからなるシーケンスを、複数のユニットに分割して記録媒体に記録し、前記ユニットは一あるいは複数のセクタから構成され、記録媒体上の各セクタの再生順を管理するセクタ管理データにより、記録媒体上に記録されたデータを管理する記録装置におけるデータ記録方法であって、各ユニットを再生すべきシーケンス上での順序を示す情報とともに、各ユニットを記録した機器を識別するための情報を各ユニットに記録することを特徴とするデータ記録方法。

【請求項3】 映像データまたは音声データからなるシーケンスを、複数のユニットに分割して記録媒体に記録し、一あるいは複数のユニットからなり、連続するセクタで構成される部分シーケンスの再生順を管理するシーケンス管理データにより、記録媒体上に記録されたデータを管理する記録媒体上に記録されたデータを管理する記録装置におけるデータ記録方法であって、

各ユニットを再生すべきシーケンス上での順序を示す情報とともに、各ユニットを記録した機器を識別するための情報を各ユニットに記録することを特徴とするデータ記録方法。

【請求項4】 映像または音声データからなるシーケンスを、複数のユニットに分割し、各ユニットに、当該ユニットを再生すべきシーケンス上での順序を示す情報と、当該ユニットを記録した機器を識別するための情報を付与して記録媒体に記録し、記録媒体の位置情報に対応させた前記ユニットの再生順を示す再生順管理データにより、記録媒体上に記録されたデータを管理する記録媒体を再生する装置におけるデータ復旧方法であって、上記再生順管理データによる再生ができない場合において、記録媒体上のユニット上で連続する2つのユニット間において、当該ユニットを再生すべきシーケンス上での順序が連続することを示し、各ユニットを記録した機器を識別するための情報が同一である場合に、2つのユニットを連続して再生するものとして、前記再生順管理データを再構築することを特徴とするデータ復旧方法。

【請求項5】 映像データまたは音声データからなるシ

ーケンスを、一あるいは複数のセクタから構成される複数のユニットに分割し、各ユニットに、当該ユニットを再生すべきシーケンス上での順序を示す情報と、当該ユニットを記録した機器を識別するための情報を付与して記録媒体に記録し、記録媒体上の各セクタの再生順を管理するセクタ管理データにより、記録媒体上に記録されたデータを管理する記録媒体を再生する装置におけるデータ復旧方法であって、

上記セクタ管理データによる再生ができない場合において、記録媒体上のユニット上で連続する2つのユニット間において、当該ユニットを再生すべきシーケンス上での順序が連続することを示し、各ユニットを記録した機器を識別するための情報が同一である場合に、2つのユニットを連続して再生するものとして、前記セクタ管理データを再構築することを特徴とするデータ復旧方法。

【請求項6】 映像データまたは音声データからなるシーケンスを、複数のユニットに分割し、各ユニットに、当該ユニットを再生すべきシーケンス上での順序を示す情報と、各ユニットを記録した機器を識別するための情報を付与して記録媒体に記録し、一あるいは複数のユニットからなり、連続するセクタで構成される部分シーケンスの再生順を管理するシーケンス管理データにより、記録媒体上に記録されたデータを管理する記録媒体を再生する装置におけるデータ復旧方法であって、上記シーケンス管理データによる再生ができない場合において、記録媒体上のユニット上で連続する2つのユニット間において、当該ユニットを再生すべきシーケンス上での順序が連続することを示し、各ユニットを記録した機器を識別するための情報が同一である場合に、2つのユニットを連続して再生するものとして、前記シーケンス管理データを再構築することを特徴とするデータ復旧方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、動画像シーケンスを記録媒体に記録する際の記録方法、さらに記録された動画像シーケンスの管理情報が正常でない場合に復旧を行う場合のデータ復旧方法に関する。

【0002】

【従来の技術】ディスクベースのビデオや音声のデジタル記録装置が普及しつつある。ディスクはテープに比べてランダムアクセスが可能という大きな利点がある。そのため、ディスクが一杯になっても、不必要なデータだけを消すことで追記可能である。また、途中で削除したり再生順序を変えることも自由にできる。これらの編集機能は、多くの場合、実際のデータを直接移動するのではなく、データを指すポインタ情報を書き換えることが実現される。なぜなら、実際のデータを移動するのに比べ、遥かに高速な実行が可能だからである。しかし、

ポインタ情報が何らかの理由で破損した場合、記録したデータを元の順序に復帰させるのは難しいという問題がある。

【0003】この問題を解決する従来技術として、特開平10-70698がある。この技術は、連続して記録したデータは、ディスク上において記録単位毎に付与されているタイムコードが連続していることを利用して、管理情報が破損した場合に、ディスクを先頭から読んでいき、記録単位毎に付与されているタイムコードを抜き出し、タイムコードの連続しているブロックを連続したデータとして扱うというものである。以下、この技術について説明する。

【0004】まず、ディスクの記録フォーマットについて説明する。図25に示すように、大きく管理領域601とデータ領域602に分けられる。管理領域602はさらにプログラムマップ603とFAT(ファイル・アロケーション・テーブル)604に分けられる。データ領域は一定のサイズ毎のブロック605、606…に分割されており、それぞれのブロックにはアクセスのための番号が付与されている。プログラムマップ603とFAT604に関しては図26に詳細を示す。プログラムマップ603はプログラム単位、つまり例えば一つの動画シーケンス単位の管理を行ない、対応する開始FATエントリやプログラム名等を保持する。FAT604はデータ領域602中で分散して記録された連続データがどのような順序で再生されるべきかを記録したものである。各エントリはデータ領域602中の1ブロックに対応する。例えば5番目のエントリは5番めのブロックに対応する。各エントリは、次のエントリへのポインタ、すなわちリンク情報を保持している。もし終了を示すコードが入っていれば、そのエントリに対応するブロックでそのデータは終了することになる。また、空きを示すコードが入っていれば、そのエントリに対応するブロックは使用されていないことがわかる。図26の場合、プログラム1はデータ領域602の0000、0001、0003ブロックにわたって記録されていることがわかる。一方、データ領域602には、ブロック毎に連続したタイムコードが時:分:秒:フレームという形式で書き込まれているとする。

【0005】ここで、管理領域601が破損した場合の処理について説明する。復旧は、次の2つの処理ステップで行なう。最初の処理ステップでは、隣接するブロックの中でタイムコードが連続しているブロックを部分プログラムとしてまとめるブロック結合処理を行う。このブロック結合処理の手順をフローチャート図27に示す。

【0006】まず、図中の記号の説明を行なう。図中のNbはディスク中のブロック数、bは現在処理の対象となっているブロック番号、Tbはブロックbに付与されたタイムコードをそれぞれ表わす。currおよびprevは現在および直前に処理の対象となったブロックに付与されたタイムコードをそれぞれ表わす。Si、Eiはi番目の部分プログラムの開始ブロック番号および終了ブロック番号

を、Bi、Fiは開始タイムコードと終了タイムコードを表わす。また、AbiはFAT604のb番目のエントリの内容を表わす。THRは正の定数で、隣接するブロックのタイムコードが連続しているかどうかの判断に用いる。Npはこのブロック結合処理で得られる部分プログラムの数を表わす。

【0007】次に、フローチャートに沿って、処理の説明を行なう。ステップ1では、ブロック0を読み込み、そのタイムコードT0を変数prevに代入する。さらに、番号0の部分プログラムを作り、その開始ブロック番号を0、開始タイムコードをT0に設定する。ステップ2では、現在の処理対象ブロック番号bを1に、処理対象部分プログラム番号を0にそれぞれ初期化する。ステップ3ではステップ4~10のループの停止条件をチェックする。ここでは、bがディスク中のブロック数Nb未満の間ループし、Nb以上になるとループを停止し、ステップ11へジャンプする。ステップ4では、現在の処理ブロックを読み込み、タイムコードTbを現在の処理対象ブロックのタイムコードを表わす変数currに代入する。ステップ5では直前のブロックのタイムコードprevと現在のブロックのタイムコードcurrが連続しているかどうかのチェックを行なう。prevとcurrの差が、ある定数THRよりも小さければ連続していると判断する。もし、連続していると判断されれば、ステップ6を、連続していないと判断されれば、ステップ7、8、9を実行する。ステップ6では、RAM103中のFATエントリAb-1にbを代入し、ブロックb-1とブロックbをリンクする。ステップ7では、現在の処理対象部分プログラムの終了ブロック番号を現在の1つ前のブロックに設定し、終了タイムコードをprevに設定する。ステップ8では、現在の処理対象部分プログラム番号を1増す。ステップ9では、現在の処理対象部分プログラムの開始ブロック番号Siをbに、開始タイムコードをcurrにそれぞれ設定する。ステップ10では、変数prevにcurrを代入し、処理対象ブロックを1増す。ステップ11では、処理対象部分プログラムの終了ブロック番号をb-1に、終了タイムコードをprevにセットする。また部分プログラムの個数Npにi+1をセットする。

【0008】第二の処理ステップは、ブロック結合処理で得られた部分プログラム間でタイムコードの連続しているものに関してつなぎあわせ、プログラム全体を再現するプログラム再現処理である。このプログラム再現処理のフローチャートを図28に示す。まず、図中の記号の説明を行なう。pは現在処理の対象となっている部分ブロック番号、qはpとの接続をチェックの対象である部分ブロック番号である。Riは、部分プログラムiがプログラムの先頭であるかどうかを示すフラグで、ここではプログラム先頭フラグと呼ぶ。その他の用いられている記号の意味は、ブロック結合処理で用いているものと同一である。次に、フローチャートに沿って、処理の説明を行なう。

【0009】ステップ1では、フラグ R_i ($0 \leq i \leq Np-1$)をtrueに初期化する。ステップ2ではインデックス用の変数 p を0に初期化する。ステップ3では、ステップ4~9のループの終了条件のチェックを行なう。 p が Np 未満であればループを継続する。ステップ4では、インデックス用の変数 q を0にセットする。ステップ5では、ステップ6~8のループの終了条件のチェックを行なう。 q が Np 未満であればループを継続する。ステップ6では、部分プログラム p と部分プログラム q の間の連続性の判定を行なう。部分プログラム p の終了タイムコードが、部分プログラム q の開始タイムコードの差が定数THRより小さければ連続していると判断され、ステップ7では、部分プログラム p 、 q を接続する。接続は以下に行なう。まず、FATエントリAepにSqを代入することで、FAT上で接続を行なう。次に、部分プログラム p の終了タイムコードおよび、終了ブロック番号を部分プログラム q のものと同じにする。最後に、部分プログラム q はプログラムの先頭にならないため、プログラム先頭フラグ R_q をfalseをセットする。ステップ2の終了時には、各プログラムの先頭に相当する部分プログラムの R_i のみがtrueになっている。したがって、プログラム先頭フラグがtrueである部分プログラムの開始ブロック番号をRAM中のプログラムマップ603に書き込めば、復帰は完了である。もし、管理領域601が物理的に破壊されているのであれば、RAM中の管理情報をディスクに書き込むことで復旧は完了する。

【0010】

【発明が解決しようとする課題】上記技術をMPEG符号化によるビデオおよびオーディオデータの記録を行なうビデオディスクレコーダに適用することを考える。MPEGビデオ規格におけるタイムコードは各ビデオプログラム毎のビデオデータの先頭からの相対時間で表される。したがって、ディスク中に複数のビデオプログラムを記録した場合、同じタイムコードを持つビデオデータが複数存在することになる。その場合、正しい順序で復元されないおそれがある。その例を図29に示す。

【0011】ここでは、ビデオプログラムA、Bがそれぞれ図29(a)のように、媒体上に分散して記録されたとする。さらにA2とB2の開始フレームのタイムコードが同じ1:00:00:00だったとする。この条件のもとで、上記の手順で内容を復元した場合、図29(b)のようにB1-A-B2となりプログラムBの間にプログラムAが挿入された形となる。なぜなら、B1の終端フレームのタイムコードが0:59:59:59でAの始端フレームのタイムコードが1:00:00:00になっていることから、上記図25で示した処理により、B1とAを連続したプログラムと判断できるためである。そのため、図29(c)に示すB1-B2-AあるいはA-B1-B2というオリジナルの提示順序が再現されない。以上の問題は、ディスク中に分散して記録された部分ビデオデータがどのビデオデータに属するのか確実に判断する手段が

ないために生じる。

【0012】一方、特開平11-162119に開示されているように、相対的なタイムコードでなく、時間情報として時：分：秒：フレームの他に記録日時が多重化されていた場合、上記の問題はある程度解決可能である。しかし、複数の機器で記録したデータがディスク中に混在していた場合、記録日時が同じデータが存在する可能性があり、同様の問題が生じるおそれがある。

【0013】本発明は、かかる従来技術の問題点に鑑みてなされたものであり、ディスク上に分散して記録する部分的なデータがどのデータに属するのか確実に判断する手段を与え、ファイルシステムが破損した場合でも、ディスク中に分散して記録されたデータの再生順序の復元を確実にすることを目的とする。

【0014】

【課題を解決するための手段】本発明にかかる記録方法は、映像や画像データを記録媒体に記録する際に、記録媒体上では複数のユニットとして分割して記録し、このユニット内に、シーケンス上での再生順序を示す情報とともに、記録した機器を識別するための情報を付与する。

【0015】このように構成することによって、再生時に、記録媒体上で、シーケンス上での再生順序を示す情報が同一のユニットが存在した場合においても、各機器にユニークに付与されている機器IDを付与しているため、シーケンスを区別することが可能である。

【0016】また、本発明にかかる再生方法では、上記したように、記録媒体上で、シーケンス上での再生順序を示す情報が同一のユニットが存在する場合を想定して、互いのユニットの順序情報が、連続するユニットを示すものであり、かつ、同一の機器ID情報を備えている場合にのみ、連続するユニットであると判断し、再生順序を示す管理情報を復旧する。

【0017】上記したシーケンス上での再生順序を示す情報は、各ユニットの再生を行う時間情報であってもよいし、各ユニットの再生順序を示す通し番号であってもよい。

【0018】

【発明の実施の形態】本発明の第1の実施形態を説明する。図1は、第1の実施形態におけるアフレコ可能なビデオディスクレコーダの構成である。図に示すように、この装置は、操作部101、CPU102、RAM103、ROM104、システムクロック105、バッファメモリ108、エンコーダ106、マルチプレクサ107、ディスクドライブ109、バス110、デマルチプレクサ111、デコーダ112、ディスク113、ECCエンコーダ/デコーダ114、IEEE-1394インタフェース115から構成される。

【0019】ディスク113は、外周から内周に向かって螺旋状に記録再生の行われる脱着可能な光ディスクとする。2048byteを1セクタとし、誤り訂正のため16セクタ

でECCブロックを構成する。ECCブロック中のデータを書き換える場合、そのデータが含まれるECCブロック全体を読み込み、誤り訂正を行い、対象のデータを書き換え、再び誤り訂正符号を付加し、ECCブロックを構成し記録媒体に記録する。

【0020】ディスク113の構成を図2に示す。ディスク中の先頭にはファイルシステム管理情報があり、その残りがファイルシステムによってファイル単位に管理されるユーザ領域となっている。ユーザ領域は管理情報領域とAVストリーム領域に分けられる。管理情報領域には管理情報に関するファイルが含まれ、AVストリーム領域には、EUS(エディタブル・ユニット・シーケンス)ファイルがある。EUSファイルは、ビデオの記録を開始してから終了するまでの一連のビデオ・オーディオデータを記録したデータストリーム単位のファイルである。以後、単にシーケンスと呼ぶ。一方、管理情報領域のファイルには、EUSファイルに関する情報を格納したEUS Managementファイルなどが含まれる。

【0021】本実施形態では、ファイルシステム管理情報によって管理されるファイルシステムを通して各ファイルのアクセスを行なう。そのため、図2中のEUSファイル#2のようにディスク中で分散して配置されたファイルを、連続した論理アドレスでアクセスすることが可能である。つまり一つのシーケンスを複数の部分シーケンスにより構成することができる。論理アドレスでのアクセスの際の単位はセクタ単位である。以下の説明におけるアドレスは特に断りがない限り論理アドレスのことを指すこととする。

【0022】記録媒体上のシーケンスの様子を図3を用いて説明する。上記したように、記録媒体上には複数のシーケンスが存在する。ここではシーケンスAとシーケンスBがある場合で、各シーケンスは所定のサイズのユニットに分割される。このユニットは記録媒体上で連続する複数のセクタにより構成されている。図3において、A4はシーケンスAの4番目のユニットであることを示している。このようなユニットの並びは再生順管理データにより管理される。上述図2のファイルシステム管理情報中が、この再生順管理データに相当する。

【0023】ファイルシステム管理情報の一例について図4を用いて説明する。図26とほぼ同様であり、ファイルエントリにファイル名を格納している。このことによって、ファイル名がわかれば、そのファイル名に対応する開始FATエントリからFATをたどって行けば、そのファイルがディスクにどのように配置されているかわかる。なお、各FATエントリは1セクタ、すなわち2048byteに対応する。

【0024】このファイルシステム管理情報は、ファイルを格納している記録媒体と同一の記録媒体に記録しておくのが一般的であるが、同一である必要はなく、一つのファイルシステム管理情報が複数の記録媒体の記録位

置を認識できるのであれば、複数の記録媒体にわたって管理してもよい。

【0025】本実施例で用いる符号化方法に関して説明する。オリジナルビデオは、MPEG-2符号化により5Mbps前後の変レートで符号化し、オーディオはオリジナル、アフレコともに、48kHzでサンプリングし、MPEG-1/LayerII符号化により2チャンネル256kbpsの固定レートで符号化する。

【0026】EUSファイルは、ビデオおよびオーディオ情報の多重化データストリームの単位であるEUSを格納するファイルである。EUSのおおまかな構成を図5に示す。EUSを構成する主要要素について、以下にまとめる。

【0027】Block:セクタに対応した2048byteの固定長の単位であり、ISO/IEC 13818-2に規定されるビデオデータおよびISO/IEC 13818-3に規定されるオーディオデータおよび他のデータを、ISO/IEC 13818-1で規定されるPES Packetにパケット化したもので構成される。

【0028】VU (Video Unit):再生時におけるランダムアクセスの単位であり、VUの先頭からアクセスすればEUSの途中であってもオーディオ、ビデオが正しくデコードされることが保証される。Blockで構成される。

【0029】PRU (Post Recording Unit):複数のVUに関連するポストレコーディングデータ(アフレコデータ)を記録するための領域である。Blockで構成される。

【0030】EU (Editable Unit):複数のVUとそれに対応する0個または1個のPRUで構成される。1つのEUはディスク中で連続的に記録する。

【0031】EUS (Editable Unit Sequence): Rec Start~StopあるいはPauseの区間に相当する単位であり、整数個のEUで構成される。

【0032】図中のblockは、2048byteの固定長の単位であり、1blockは1セクタに格納される。1個のblockは原則として1個のパケットで構成される。ここでのパケットは、ISO/IEC 13818-1で規定されるPES packetに準拠する。パケットの構成を図6に示す。パケットは、そのパケットに関する属性等を格納するパケットヘッダとビデオデータ等の実際のデータを格納するパケットデータで構成される。パケットヘッダに含まれる主な情報は以下の通りである。packet-start-code-prefixはISO/IEC 13818-1で規定されたパケットの開始コードである。stream-idはこのパケットの種類を表わす。PES-packet-lengthはこのフィールド以降のデータのサイズを表わす。PES-header-data-lengthはパケットヘッダのサイズを表わす。PTS(プレゼンテーション・タイム・スタンプ)は、多重化したオーディオやビデオといったエレメンタリ・ストリーム間の同期情報であり、パケット中に先頭が含まれるアクセスユニット(ビデオの場合1フレーム)が再生されるタイミングを90kHzのクロックでカウントした値を33ビットで表わしたものである。DTS(デコー

ディング・タイム・スタンプ)は、そのパケット中に先頭があるアクセスユニットがデコードされるタイミングをPTSと同じ時間軸で表わしたものである。stuffing-bytesは、次に説明するようにパケットのサイズを調整するために用いられる。

【0033】もし、パケットが2048byteに満たず、不足分が7byte未満のときはパケットヘッダにスタフピング・バイトを入れる。一方不足分が7byte以上のときは不足分に相当するパディングパケットをそのパケットの後に置く。このスタフピング・バイト、パディングパケットは実際に処理を行わないいわゆるダミーデータである。本実施例で用いるパケットを以下にまとめる。

【0034】V-PKT (Video Packet): ISO/IEC 13818-2で規定されるビデオデータを格納したパケット

A-PKT (Audio Packet): ISO/IEC 13818-3で規定されるオーディオデータを格納したパケット

P-PKT (Padding Packet): ISO/IEC 13818-1で規定されるパディング用パケット

UH-PKT (Unit Header Packet): VU、PRUに関するヘッダを格納したパケット

V-PKT、A-PKTおよびP-PKTのフォーマットはISO/IEC 13818-1の規定に準拠する。その他のパケットのフォーマットについては後述する。また、EUSを構成するblockを以下にまとめる。

【0035】V-BLK (Video Block): V-PKTを格納したblock

A-BLK (Audio Block): A-PKTを格納したblock

P-BLK (Padding Block): P-PKTを格納したblock

UH-BLK (Unit Header Block): UH-PKTを格納したblock、EUについて説明する。EUの構造を図7に示す。EUは1個以上の整数個のVUと0個または1個のPRUを含む。1個のEUSを構成するVUの提示時間は同一にする。ただし、EUSの最後のVUは他のVUより短くてもよい。つまり、1つのEUSにおけるVUの再生間隔は常に同一となっている。なお、VUの提示時間は、そのVUがビデオデータを含む場合は、そのVUに含まれるビデオフィールド数あるいはビデオフレーム数にそれぞれビデオフィールド周期あるいはビデオフレーム周期をかけたものとして定義する。

【0036】1個のEUSを構成するEUは、すべてPRUを含むか、すべてPRUを含まないかのいずれかにする。EUを構成するVUの個数Nvuは、EUSの最後のEUを除きEUS内では一定にする。つまり、1つのEUSにおいて、EUの提示時間間隔は常に一定となる。PRUを持たないEUSの場合、Nvu=1とする。一方、PRUを持つEUSの場合、VUあたりの提示時間をTpv、回転待ち時間をTv、現在読込中のトラックからアフレコ領域のあるトラックへジャンプする時間をTk、ディスクからのデータ転送速度をRs、EUS全体のビットレートをRo、アフレコ音声のチャンネルあたりのビットレートをRa、アフレコ音声のチャンネル数をNchとしたとき、

【0037】

【数1】

$$\text{ceiling}\left(\frac{2 \times (T_k + T_v) \cdot R_s}{(R_s - R_o - R_a \cdot N_{ch}) T_{pv}}\right) \leq N_{vu} \leq \text{floor}\left(\frac{10 \text{sec.}}{T_{pv}}\right)$$

とする。なお、ceiling(x)はx以上の最小の整数を、floor(x)はx以下の最大の整数を求める関数である。PRUを持つEUSの場合にデータ転送速度などに基きNvuの最小値を設定する理由は、EUあたりの時間が十分に大きくなると、逐次的にアフレコを行なう際、ヘッドを現在の読込位置からアフレコ領域へ移動させるオーバーヘッドの占める割合が大きくなり、データの読込が表示に追いつかなくなりビデオやオーディオの再生が途切れてしまうからである。

【0038】次にVUについて説明を行なう。VUは、sequence-headerおよびそれに続くGOP-headerを直前に置いた1個以上の整数個のGOP(グループ・オブ・ピクチャ)からなるビデオデータと、それと同期する整数個のAAU(オーディオ・アクセス・ユニット)からなるオーディオデータを含む。GOPは、MPEGビデオ圧縮の単位で、複数のフィールド群あるいはフレーム群で構成される。AAUは、オーディオサンプルを0.024秒毎にセグメント化しそれぞれのセグメントを圧縮したものである。GOP、AAUともにそれぞれの単位の先頭からデコードする必要があるが、VUはそれぞれを整数個含んでいるためVU単位で独

立再生可能である。1VUあたりのビデオフィールド数はNTSCの場合、24フィールドから60フィールド、PALの場合には20フィールドから50フィールドの範囲にする。

【0039】VUは図8に示すように、先頭にUnit Header Block(UH-BLK)、次に前述のオーディオデータを格納したA-BLKの列を置き、最後に前述のビデオデータを格納したV-BLKの列の順に配置する。A-BLKの個数は、前述のオーディオデータを格納するのに必要十分なものにす。最後のA-BLKに余りが出た場合には前述のようにP-PKTあるいはスタフピングバイトで調整する。V-BLKも同様の構成とする。

【0040】次にPRUについて説明を行なう。PRUは、1以上整数個のVUに対するオーディオを格納するための領域であり、1個のEUに0個あるいは1個存在する。PRUのサイズは、EUあたりの提示時間に対応するオーディオデータとPRUヘッダ・ブロックを含むことのできる最小の整数個のECCブロックである。PRUを構成するECCブロックの数NPRU、ECCは

【0041】

【数2】

$$N_{PRU,ECC} = \text{ceiling}((1 + \text{ceiling}(\frac{Ra \cdot Nch \cdot Tpv}{2048 - 14}) \times Nvu) / 16)$$

として規定される。なお、PRU中に記録するオーディオデータは、そのPRUが含まれるEU中のVUのオーディオと同じデータレート、同じサンプリング周波数で記録する。

【0042】オリジナルデータ記録直後のPRUの構成を図9に示す。先頭にUnit Header Block (UH-BLK)を1個記録し、残りの領域をPadding Block (P-BLK)で埋めておく。つまり、オリジナルデータ記録直後の時点では、オーディオデータは記録されていない。

【0043】PRUにオーディオをアフレコした後のPRUの構成を図10に示す。先頭にUnit Header Block (UH-BLK)を1個記録し、その後にはそのEUに同期したオーディオデータをA-BLKの列として記録し、残りの領域をPadding Data (P-BLK)で埋めておく。このとき、PRU中のA-BLKは、同じEU中のそれぞれのVUに含まれるA-BLK数の合計と同じ数にする。さらに、PRU中のそれぞれのA-BLKの持つPTSの値が同EU中のそれぞれのVUに含まれるA-BLKのPTSと同じ順番でかつ、同じ値を取るように、ポストレコーディングのオーディオデータを記録する。すなわち、アフレコ後PRU中には、各VUに含まれるA-BLKの列に対応するA-BLKの列が存在することになる。このような、VUに対応したPRU中のA-BLKの列をSAU(サブ・オーディオ・ユニット)と呼ぶことにする。なお、言うまでもないが、SAUには、VUと同様整数個のAAUが含まれることになる。

【0044】EU中でのPRUの配置について説明する。PRUは、それが含まれるEUの先頭の15セクタ以内のECC境界、つまりEU中の最初に現れるECC境界に置く。例えば、あるEUの先頭がECCブロック境界だった場合、図11(a)のように、そのEUの先頭の直後にPRUを配置する。また、EUの先頭がECCブロック境界でなかった場合は、(b)のように、EUの境界の直後から15論理ブロック以内のECCブロック境界、つまりEU中の最初に現れるECCブロック境界に配置する。この場合、EU中の先頭のVUはPRUによって分断されることになる。

【0045】UH-PKTの構造を図12に示す。図中のBP(バイト・ポジション)は先頭からの相対的なバイト位置であり、バイト数はそれぞれのフィールドのバイト数を示す。packet-start-code-prefix、stream-id、PES-packet-lengthは前述の通りである。Unit Propertyは1byteのビットフィールドで、このUnit headerが含まれるユニット (PRUおよびVU) に関する情報を格納する。その中にはフィールドFirst VU of EUおよびフィールドRecorder ID Formatが存在する。フィールドFirst VU of EUはそのUH-PKTを含むユニットがEU中の先頭のVUであれば1それ以外は0に設定される。フィールドRecorder ID Formatは2ビットのフィールドであり、01b(xxbはxxが2進数であることを示す)の場合、MAC(メディア・アクセス・コ

ントロール)アドレス(イーサネットにおける機器ID)、10bの場合、GUID(グローバル・ユニーク・ID: IEEE 1394における装置ID)をフィールドRecorder IDに格納することを意味する。フィールドRecorder IDは、そのユニットを記録した装置のIDを表す。GPSTimeStamp Offsetは、そのユニットが含まれるEUSの記録を開始した日時を格納するフィールドであり、1980年1月6日00:00(UTC: Universal Time Coordinated)からの秒数を32ビットの符号無し整数で表す。Unit Start PTは、そのUH-PKTが含まれるユニットの先頭のV-PKT中のPTS値の最上位ビットを省略したものである。なお、以後このようにPTSの最上位ビットを省略した形式をPTフォーマットと呼ぶことにする。Length of UnitはこのUnit headerが含まれるユニット中のblock数を表わす。Start RLBN of Video Dataは、ユニットの先頭からビデオデータが始まるまでのblock数を表わす。

【0046】UH-PKTは、他のデータとは独立したセクタの先頭に配置され、しかもパケットの先頭がpacket-start-code-prefix、stream-id(0000 0000 0000 0000 0000 0001 1011 1101b)という特定のビットパターンを持つため、ファイルシステムが破損した場合でもセクタを順に読んで行けば容易に検出可能である。

【0047】また、時間情報等、ユニット間の関係を示す情報をAVデータとは独立のパケット(UH-PKT)に置いてあるため、AVデータをデコードすることなく、ユニット間の関係を判断することが可能である。

【0048】また、UH-PKT中にそのEUSを記録を開始した時刻(GPSTimeStamp Offset)を記録してあるため、別の日の同じ時刻に記録したユニット同士を区別可能である。さらに、UH-PKT中にそのユニットを記録した装置のID(Recorder ID)を記録しておくことで、異なる機械によって同じ日の同じ時刻に記録されたユニットが同一ディスク内に存在した場合にも区別が可能である。なぜなら、Recorder IDに記録するMACアドレスやGUIDは、世界に1つしかない、その記録装置固有(正確に言えばネットワークインタフェース部固有)の値であるため、異なる機械で記録したユニットには必ず異なるRecorder IDが付加されることになるためである。つまり、このフォーマットで記録した部分ビデオデータ(VUおよびPRU)は、同一のビデオデータ(EUS)を構成するものであれば、Recorder IDとGPSTimeStamp Offsetの両方が必ず一致し、異なるビデオデータに属するものであれば、少なくとも一方が必ず異なることになる。

【0049】したがって、従来技術では復旧が困難である図29のようなビデオデータ配置の際にも、B1とAが異なるビデオデータを構成していれば、図13に示すように、B1-A間でRecorder IDあるいはGPS TimeStampOffset(図ではGPS TSO)のいずれかが必ず異なるため、B1とAは

異なるビデオデータの部分ビデオデータであると容易に正しく判断することが可能である。

【0050】EUS Managementファイルの構造を図14に示す。EUS Managementファイルは、ディスク中に記録されたすべてのEUSファイルを管理するための情報を格納したものである。以下、本実施形態の説明に必須な項目のみについて説明を行なう。フィールドNumber of EUSIは、このファイルで管理するEUSファイルの個数を表わす。フィールドEUSI (EUS Information)は各EUSファイルに関する情報であり、Number of EUSI個分存在する。EUSIはさらに図15のように構成される。図中のStart PTおよびEnd PTは、このEUSIが管理するEUSファイル中の開始PTSおよび終了PTSの最上位ビットを省略したものである。なお、以後このようにPTSの最上位ビットを省略した形式をPTフォーマットと呼ぶことにする。Post Recording UnitSizeはこのEUSIが管理するEUSファイル中のPRUのサイズを表わす。

【0051】Address LUT(ルックアップ・テーブル)は、PTフォーマットで記述されたタイムコードからそのタイムコードに対応するデータが記録されているアドレスを検索するためのテーブルである。Address LUTの構成を図16に示す。フィールドPBTime of EUは、EUあたりの提示時間を1/90000[秒]単位で表わしたものであり、PTフォーマットと同じスケールとなっている。PB Time of VUも、同様にVUあたりの提示時間を1/90000[秒]単位で表わしたものである。Number of PRU InformationはAddress LUT中のPRU Informationの数であると同時に、EUS中のPRUの個数も表わす。Number of VU Informationも同様にAddress LUT中のVU Informationの数およびEUS中のVU数を表わしている。

【0052】図17は、PRU Informationの内容を表わす。図中のRLBN of PRUはそのPRU Informationが管理するPRUのアドレスを表わす。図18は、VU Informationの内容を表わす。図中のRLBN of VUはそのVU Informationが管理するVUのアドレスを表わす。

【0053】Address LUTを用いて、あるタイムコードPTに対応するPRUのアドレスを求める手順を以下に示す。まず、PTからEUSI中のStartPTを引くことで相対PTを求め、次に相対PTをPB Time of EUで割り、小数部を切り捨てることで、そのPTに対応するPRUを管理するPRU Informationのインデックスが求まる。次に、そのインデックスに対応するPRU Information中のRLBN of PRUで与えられるアドレスが、目的とするPTに対応するPRUのアドレスである。時刻PTに対応するVUのアドレスも同様に、PTからStart PTを引いたものをPB Time of VUで割り、小数部を切り捨てた値に対応するインデックスのVU Information中のRLBN of VUを参照することで得られる。このように単純な処理でVUやPRUの先頭アドレスが得られるのは、EUおよびVUあたりの提示時間を一定にしているためである。

【0054】上記ディスクフォーマットで記録を行なう際の手順を以下に示す。なお、以下の説明ではビデオはNTSCで記録し、VUを30フィールドからなる1個のGOPで構成し、ビデオ最大ビットレートを8[Mbps]とする。ディスク転送レートRsは12[Mbps]、アフレコ領域への最大ジャンプ時間Tkを0.3[秒]、最大回転待ち時間Tvを0.2[秒]とする。また、オーディオビットレートおよびオーディオチャンネル数をそれぞれ0.125[Mbps/チャンネル]、2[チャンネル]とし、オリジナルおよびアフレコで共通に用いることにする。このとき、VUあたりの提示時間Tpvは上記のGOP構成により約0.5秒となる。また、アフレコが可能なEUあたりのVU数Nvuの範囲は、 $7 \leq Nvu \leq 20$ となる。本実施形態では、 $Nvu=8$ 、すなわちEUあたりの提示時間は約4秒となる。

【0055】オリジナルプログラム記録時のCPU102の処理の流れを図19に沿って説明する。すでに、ディスクからEUS Managementファイルやファイルシステム管理情報がRAM103に読み込まれているものとする。CPU102はエンコーダ106を起動し、次にファイルシステム管理情報を基に1EU分のデータを記録するのに十分な連続領域がディスク上にあるかどうか調べる(ステップ1)。もし、なければ録画を停止する。

【0056】もし十分な領域があれば、記録対象のVUがEU中の何番目のVUかを表わす変数iを0にリセットし、空き領域先頭アドレスを変数addrに記憶させる(ステップ2)。次に、マルチプレクサ107から1VU分のデータがバッファメモリ108にバッファリングされたことの通知を待つ(ステップ3)。マルチプレクサ107から通知が来たら、変数iが0のときは、変数addrがECCブロック境界かどうかを判断し(ステップ9)、もし、ECCブロック境界でなければ、次のECCブロック境界までバッファメモリ108中のVUデータをディスクに記録する(ステップ10)。次に、UH-PKTおよびP-PKTでPRUをRAM103中に構成し、それをディスクに記録する(ステップ11)。次に、バッファメモリ108中の先頭のVUデータをディスクに記録する(ステップ6)。記録が終わったら変数iをインクリメントする(ステップ7)。変数iがEU中のVU数を表わす変数Nvuより小さければステップ4にジャンプし(ステップ8)、等しくなればステップ2にジャンプする。以上の処理を、操作部101から停止指令がきたり、ディスク中に十分な連続領域が無くなるまで、EU単位に行なっていく。

【0057】以上のCPU102の処理と並行して、マルチプレクサ107は、オーディオ、ビデオそれぞれのエンコーダ106から送られるデータにPTS等を付与しバケット化しバッファメモリ108に貯えていく。また、CPU102から記録開始日時およびIEEE-1394インタフェース115の持つGUIDを受け取り、UH-PKTを構成する。1GOP分のV-PKTとそれに同期するA-PKTがバッファメモリ108に貯えられたらCPU102にVU分のデータをバッファリングしたことを通知する。

【0058】ここで、管理領域602が破損した場合の処理について説明する。復旧は、ディスクを先頭から順に読みこんでいき、UH-BLKを目印にVUやPRUの検出をする(ユニット検出処理)、VU間の順番を復元する(ユニット再生順復元処理)、復元されたVUの再生順を基にファイルシステムを復元する(ファイルシステム復元処理)、の順に行なう。

【0059】ユニット検出処理のフローを図20に示す。まず、図中の記号の説明を行なう。bは処理の対象となっているセクタの番号を表す。NvとNpは、検出済みのVUおよびPRUの個数をそれぞれ表す。Sv[i]、Ev[i]、Iv[i]、Ov[i]、Pv[i]、Lv[i]はそれぞれ、i番目に検出されたVUの開始セクタ番号、終了セクタ番号、Recorder ID、GPSTimeStamp Offset、Unit Start PT、Unit Lengthをそれぞれ表す。一方、Sp[j]、Ep[j]、Ip[j]、Op[j]、Pp[j]、Lp[j]はそれぞれ、j番目に検出されたPRUの開始セクタ番号、終了セクタ番号、Recorder ID、GPS TimeStamp Offset、Unit Start PT、Unit Lengthをそれぞれ表す。

【0060】次にフローチャートに沿ってユニット検出処理の説明を行なう。まず、b、Nv、Npをそれぞれ0にセットする(ステップ1)。次に、UH-BLKを読みこむまで、セクタを読み飛ばす(ステップ2～ステップ4)。UH-BLKかどうかは、読みこんだセクタの先頭が特定のビット列(packet-start-code-prefix, stream-id)かどうかで判断する。次に、そのUH-BLKが管理するユニットがPRUかVUかを判断する(ステップ4)。ここではStart RLBN of Video Dataが0でなければVUと判断する。もしVUと判断した場合ステップ6にジャンプし、PRUと判断した場合ステップ8にジャンプする。ステップ6ではUH-BLKの各種フィールドのうち、Recorder ID、GPSTimeStamp Offset、Unit Start PT、Unit Lengthの値をNv番目のVUに関する変数であるIv[i]、Ov[i]、Pv[i]、Lv[i]にそれぞれ格納する。ステップ7ではNvをインクリメントする。ステップ8ではUH-BLKの各種フィールドの値をNp番目のPRUに関する各種変数に格納する。ステップ9ではNpをインクリメントする。ステップ10ではbをインクリメントし、ステップ2にジャンプする。以上の処理により、ディスクに記録されたすべてのVUおよびPRUの情報が得られる。次に、これらの情報を用いて、同じEUを構成するVUとPRUを結びつける。処理のフローチャートを図21に示す。まず、新たに追加した記号の説明を行なう。Xv[j]はj番目のVUに対応するPRUの番号を表す。値が-1の場合、対応するPRUは無いことを表す。

【0061】次にフローチャートに沿って処理の説明を行なう。まず、変数の初期化を行なう(ステップ1)。ステップ2～8はPRUを指すインデックスiをインクリメントしiがNp未満の間繰り返す。VUを指すインデックスjを初期化し、ステップ4～7は、jがNv未満の間繰り返す。ステップ5ではi番目のPRUとj番目のVUの変数のうち、Reco-

order ID、GPSTimeStamp Offset、Unit Start PTを示す変数をそれぞれ比較し、すべてが一致すればXv[j]にiを代入する。以上の処理により、j番目のVUと同じタイミングに再生されるPRUを、Xv[j]の値を参照することでわかるようになる。

【0062】次に、ユニット再生順復元処理のフローを図22に示す。新たに定義した記号の説明を行なう。next[i]にはi番目のVUに後続するVUの番号を格納するものであり、値が-1であれば、後続するVUが存在しないことを示す。top[i]は、i番目のVUがシーケンス(EUS)の先頭かどうかのフラグであり、trueなら先頭、falseなら非先頭を意味する。p、qは、現在、連続性の判断の対象となっている2つのVUのうち先行するVUおよび後続のVUの番号をそれぞれ表す。THRは、VUの連続性を判断するための閾値であり、本実施例ではVUあたりの提示時間の上限である1秒を1/90000[秒]単位で表わしたもの、すなわち90000とする。

【0063】次にフローチャートに沿ってユニット再生順復元処理の説明を行なう。ステップ1ではnext[i]およびtop[i]の初期化を行ない、それぞれのVUには後続するVUがなく、しかもEUSの先頭に位置しているものとみなす。ステップ2では、pを初期化する。ステップ3では、ステップ4～9の終了条件をチェックする。ステップ4では、qを初期化する。ステップ5では、ステップ6～8の終了条件をチェックする。ステップ6では、p番目のVUにq番目のVUが後続しているかどうかのチェックを行なう。Ov[p]とOv[q]、Iv[p]とIv[q]がそれぞれ一致し、Pv[p]がPv[q]より小さく、しかもPv[q]はPv[p]+THR以下である場合に、p番目のVUにq番目のVUが後続していると判断し、ステップ7にジャンプする。このステップ6の条件を満たさない場合、ステップ8に処理をすすめる。

【0064】ステップ7では、p番目のVUにq番目のVUが後続していることを表すため、next[p]にqを格納する。さらに、q番目のVUはEUSの先頭ではないため、top[q]にfalseを格納する。以上の処理により、変数top[]がtrueのVUを開始VUとして、next[]をたどっていけば、VUの正しい再生順が得られることになる。

【0065】ファイルシステム復元処理では、前述のユニット再生順復元処理で得られたVUの連続情報から、ファイルシステムの復旧を行なう。ファイルシステム復旧のフローを図23に示す。新たに定義した記号について説明する。file[i]は、ファイルエントリ中のi番目のファイルの開始FATエントリを表す。A[b]はb番目のFATエントリ中の値を表す。

【0066】次にフローチャートに沿ってファイルシステム復元処理の説明を行なう。まず、VUのインデックス変数pおよび、ファイルエントリのインデックス変数iの初期化を行なう(ステップ1)。ステップ2では、ステップ3～14までの処理の終了条件をチェックする。ステップ3では、p番目のVUがEUSの先頭かどうかのチェックを行な

う。もし先頭ならステップ4にジャンプし、そうでなければステップ13へスキップする。ステップ4では、VUの連鎖をたどるための変数である q に p の値を格納する。ステップ5では、 $file[i]$ に q 番目のVUの開始セクタ番号を代入する。ただし、 $Xv[q]$ が0より大きくしかも $Sp[Xv[q]]$ の方が $Sv[q]$ より小さい場合、図11(a)のように配置されていると判断し、 $file[i]$ には $Sp[Xv[q]]$ をセットする。ステップ6では、 q 番目のVUの開始セクタ番号 $Sv[q]$ およびVUのセクタ数 $Lv[q]$ を変数 b および変数 g にそれぞれ代入する。ただし、 $Xv[q]$ が0以上、すなわち対応するPRUを持つ場合であって、 $Sp[Xv[q]]$ の方が $Sv[q]$ より小さい場合、PRUが図11(a)のように配置されていると判断し、 b には $Sp[Xv[q]]$ をセットする。また、そのPRUのサイズを含めるために、 g には $Lv[q]+Lp[Xv[q]]$ を代入する。 $Sp[Xv[q]]$ の方が $Sv[q]$ より大きい場合、PRUは図11(b)のように配置されていると判断し、 b には $Sv[q]$ を、 g には $Lv[q]+Lp[Xv[q]]$ をセットする。ステップ7〜9は、 g が0より大きい間、 $A[b]$ への $b+1$ の格納(ステップ8)、 b のインクリメント、 g のデクリメント(ステップ9)を繰り返す。この繰り返しにより、VU内(あるいはEUSがPRUを持つ場合には、EUSの先頭のVUはPRUを含む)に関するFATの修復を行なうことになる。ステップ7で、 g が0になった場合、ステップ10にジャンプする。ステップ10では、後続するVUの番号 $next[q]$ を q に格納する。 q が-1、すなわち後続するVUがなければ、ステップ12にジャンプし、そうでなければ、ステップ14にジャンプする。ステップ14では、 $A[b]$ に q 番目のVUの開始セクタ番号を格納して、ステップ6にジャンプする。ただし、このときも、ステップ6と同様、 $Xv[q]$ が0以上の場合、すなわち対応するPRUが存在する場合は、そのPRUの開始セクタ番号 $Sp[Xv[q]]$ をチェックし、小さい方の値を $A[b]$ に格納する。ステップ12では、 $A[b]$ に終了を意味する値を格納し、 i をインクリメントする。以上の処理がすべて終了した後、RAM103上の $A[b]$ 、 $file[i]$ をディスク113中のFAT、ファイルエントリにそれぞれ記録する。その際、ファイルエントリ中の各ファイル名には他のファイルと重複しない名前を与える。以上の処理により、各EUSを通常のファイルとしてアクセスすることが可能になり、EUSを先頭からファイルシステムが与える順序でセクタを読みこんでいけば記録したときの順序で再生可能になる。

【0067】本実施例ではシーケンス再生管理情報としてFATを用いているが、図24に示すように、EUSファイル(シーケンス)を構成する連続領域(部分シーケンス)に関する情報を用いることも可能である。この場合、部分シーケンスに関する情報はその部分シーケンスの開始セクタ番号及びその連続領域のセクタ数で構成する(開始と終了セクタでもよい)。このような構成により、FATと同様、ファイルを構成するセクタの再生順を管理することができる。なお、各部分シーケンスは1個以上の

ユニットで構成されていることは言うまでもない。このように、本発明における再生管理順情報は、FATに限定されるものでなく、図3に示すように、ユニットの再生順を管理する情報であれば、どのような管理方式であってもよい。

【0068】本実施例では、UH-PKT中にUnit Start PTを記録し、その値を用いてユニット間の連続性の判断を行なっているが、代わりにそのEUSの先頭を0とした通し番号を用いても構わない。つまり、ユニット間の連続性(順序)を表す情報であれば、どのようなものであっても、復旧は可能である。

【0069】本実施例では、UH-PKTを置く間隔を数秒単位にしているが、同様な情報を置く間隔をもっと長い単位や、もっと短い単位(例えばセクタ単位)に設定することも可能である。また、本実施例では、VUは同一の再生時間であるが、連続性を示す情報をおく間隔は、時間単位でなく、データ単位であってもよい。さらに、同一間隔である必要もない。ただし、あるUH-PKTから次のUH-PKTの直前までの領域は連続的に記録する必要があるため、あまり間隔を長くすると、ディスク中の空き領域の使用に制限が大きくなる。一方、あまり間隔を短くすると、無駄が大きくなる。

【0070】本実施例では、UH-PKTにそのユニットの記録を行なった装置のIDを記録したが、以下のようなバリエーションも考えられる。

【0071】装置の種類によっては、IEEE-1394やイーサネット等のネットワーク機能を装備しておらず、その装置の持つユニークなIDがない場合がある。その場合、UH-PKTに、そのユニットがその装置で記録したものか他からコピーしたものかを区別するための情報を設けておくことが考えられる。この場合、他の複数の装置で記録されたユニット間の区別はできないが、他の装置で記録したユニットとその装置で記録したユニットの区別は可能であるため、その装置で記録したデータだけは確実に復旧可能である。

【0072】また、上記の構成では、複数のテレビ番組を同時にディスクに記録する装置の場合、同じ装置ID、時刻情報を持つユニットが存在してしまう。そのような場合には、番組を区別するための情報、チャンネル番号や番組のID等をUH-PKTに記録すればよい。

【0073】また、上記の構成では、複数のカメラから入力されるビデオを同時にディスクに記録する装置の場合、第1のバリエーションと同様、同じ装置ID、時刻情報を持つユニットが存在してしまう。そのような場合、UH-PKTにカメラのIDあるいは、入力チャネルのIDを記録すればよい。

【0074】このように、本発明における「機器を識別するための情報」は特に装置特有に備えられているIDに限るものではなく、装置内における入力チャネルなど、データの入手先を特定することができる情報であればよ

い。

【0075】

【発明の効果】本発明によれば、再生時に、記録媒体上で、シーケンス上での再生順序を示す情報が同一のユニットが存在した場合においても、各機器にユニークに付与されている機器IDを付与しているため、シーケンスを区別することが可能である。このことにより、記録媒体上で、ユニットが分散して記録されており、この分散されたユニットの順序を管理しているファイルシステムが破損した場合でも、管理情報を復旧して、確実にシーケンスを再生することが可能となる。

【図面の簡単な説明】

【図1】本発明の一実施形態における構成を示すブロック図である。

【図2】本発明の一実施形態におけるディスク中のデータ配置である。

【図3】本発明の一実施形態における再生順管理データを示す図である。

【図4】本発明の一実施形態におけるファイルシステム管理情報を示す図である。

【図5】本発明の一実施形態におけるEUSファイルの概要を示す図である。

【図6】パケットの構造を示す図である。

【図7】本発明の一実施形態におけるEUの構造を示す図である。

【図8】本発明の一実施形態におけるVUの構造を示す図である。

【図9】本発明の一実施形態におけるアフレコ前のPRUの構造を示す図である。

【図10】本発明の一実施形態におけるアフレコ後のPRUの構造を示す図である。

【図11】本発明の一実施形態におけるPRUの配置に関する図である。

【図12】本発明の一実施形態におけるUH-PKTの配置に関する図である。

【図13】本発明の一実施形態における記録媒体上でのシーケンスを示す図である。

【図14】本発明の一実施形態におけるEUS Managementファイルの構造を示す図である。

【図15】本発明の一実施形態におけるEUSIの構造を示す図である。

【図16】本発明の一実施形態におけるAddress LUTの構造を示す図である。

【図17】本発明の一実施形態におけるAddress LUT中のPRU Informationの構造を示す図である。

【図18】本発明の一実施形態におけるVU Informationの構造を示す図である。

【図19】本発明の一実施形態におけるオリジナルデータ記録のフローチャートである。

【図20】本発明の一実施形態におけるデータ復旧のユニット検出処理の前半部分を示すフローチャートである。

【図21】本発明の一実施形態におけるデータ復旧のユニット検出処理の後半部分を示すフローチャートである。

【図22】本発明の一実施形態におけるデータ復旧のユニット再生復元処理を示すフローチャートである。

【図23】本発明の一実施形態におけるデータ復旧のファイルシステム復元処理を示すフローチャートである。

【図24】本発明の一実施形態における部分シーケンスにより再生順を管理するシーケンス管理データを示す図である。

【図25】従来のディスクフォーマットを示す説明図である。

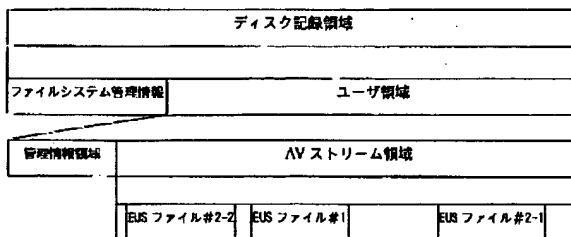
【図26】従来の管理情報を示す説明図である。

【図27】従来のディスク記録方法における管理情報復旧手順におけるブロック結合処理を示すフローチャートである。

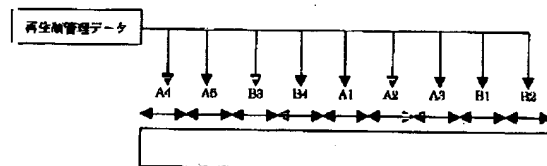
【図28】従来のディスク記録方法における管理情報復旧手順におけるプログラム再現処理を示すフローチャートである。

【図29】従来のディスク記録方法における管理情報復旧後のデータ再生順を示す説明図である。

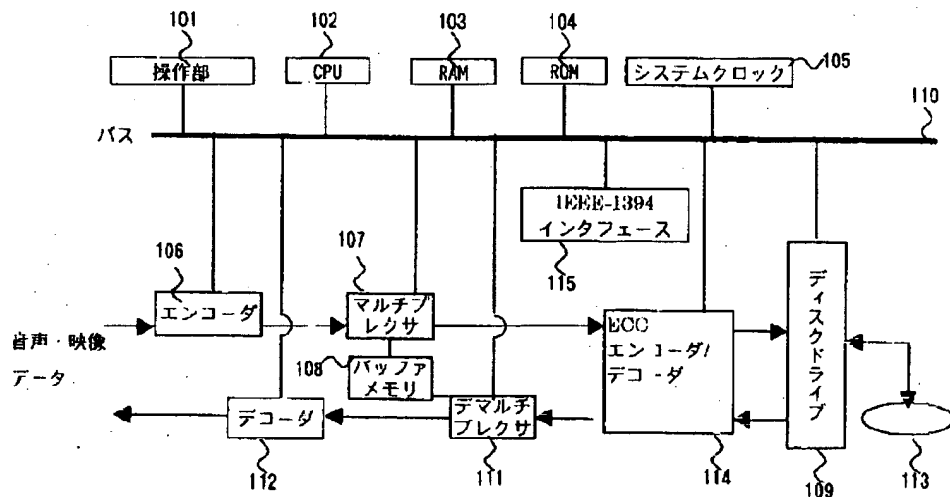
【図2】



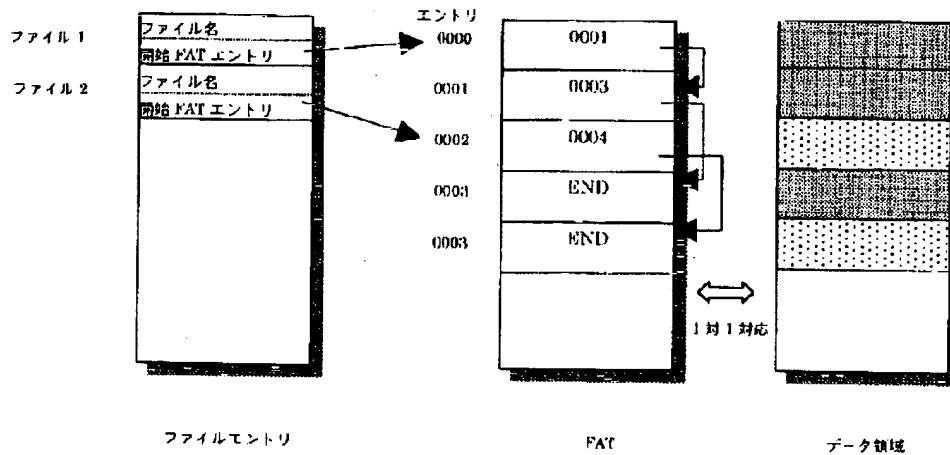
【図3】



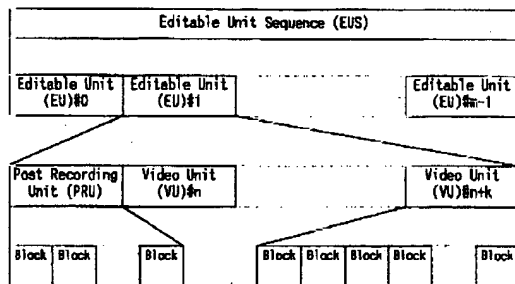
【図1】



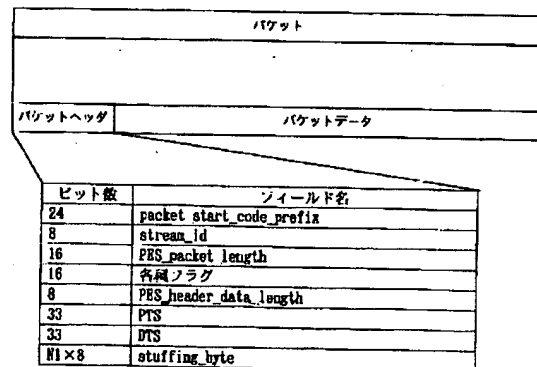
【図4】



【図5】



【図6】



【図15】

BP	バイト数	フィールド名
0	4	EUSI ID
4	4	EUSI Size
8	23	Title Text
31	1	Character Code
32	6	Time Stamp Creation
44	6	TimeStamp-Modification
50	10	Text Information
60	10	Thumbnail Information
70	2	Data File ID
72	4	Data File Size
76	4	Start PT
80	4	End PT
84	2	EUS Property
88	2	Video Property
90	4	Camera Property
-	2	Audio Property
-	2	Post Recording Unit Size
-	2	Post Recording Property
-	64	Source Information
-	64	Copyright Property
-	2	Number of Still Picture
-	-	Still Picture Information
-	-	Address LUT
-	-	Reference Information

【図16】

BP	バイト数	フィールド名
0	4	Address Offset
4	4	PB Time of EU
8	4	PB Time of VU
12	4	Number of PRU Information (=NOPI)
16	4	Number of VU Information (NOVI)
20	4×NOPI	PRU Information
-	n×NOVI	VU Information

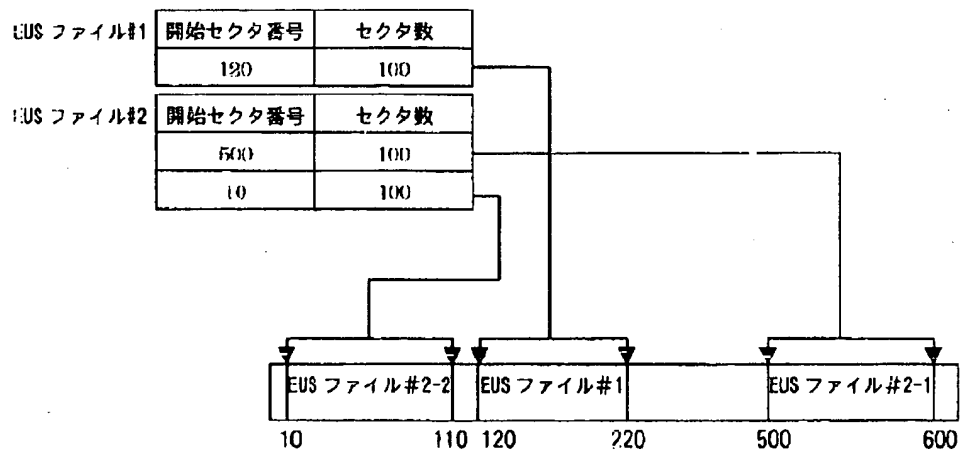
【図18】

BP	バイト数	フィールド名
0	3	RLBN of VU
3	1	VU Status
4	1	Number of IP Pictures
5	2×NOIP	End RLBN of IP Pictures

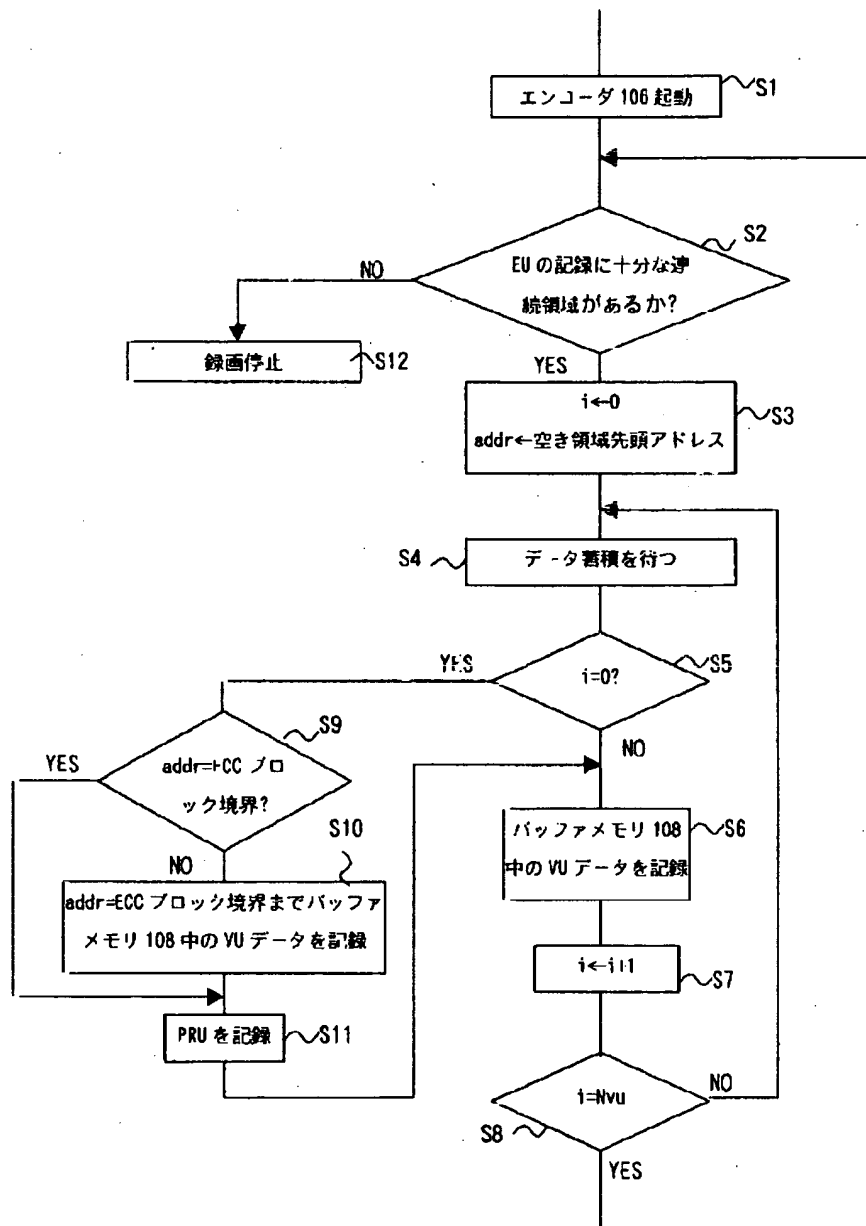
【図17】

BP	バイト数	フィールド名
0	3	RLBN of PRU
3	1	PRU Status

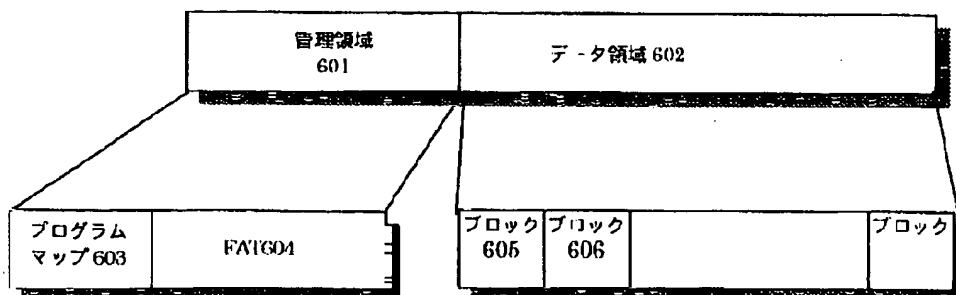
【図24】



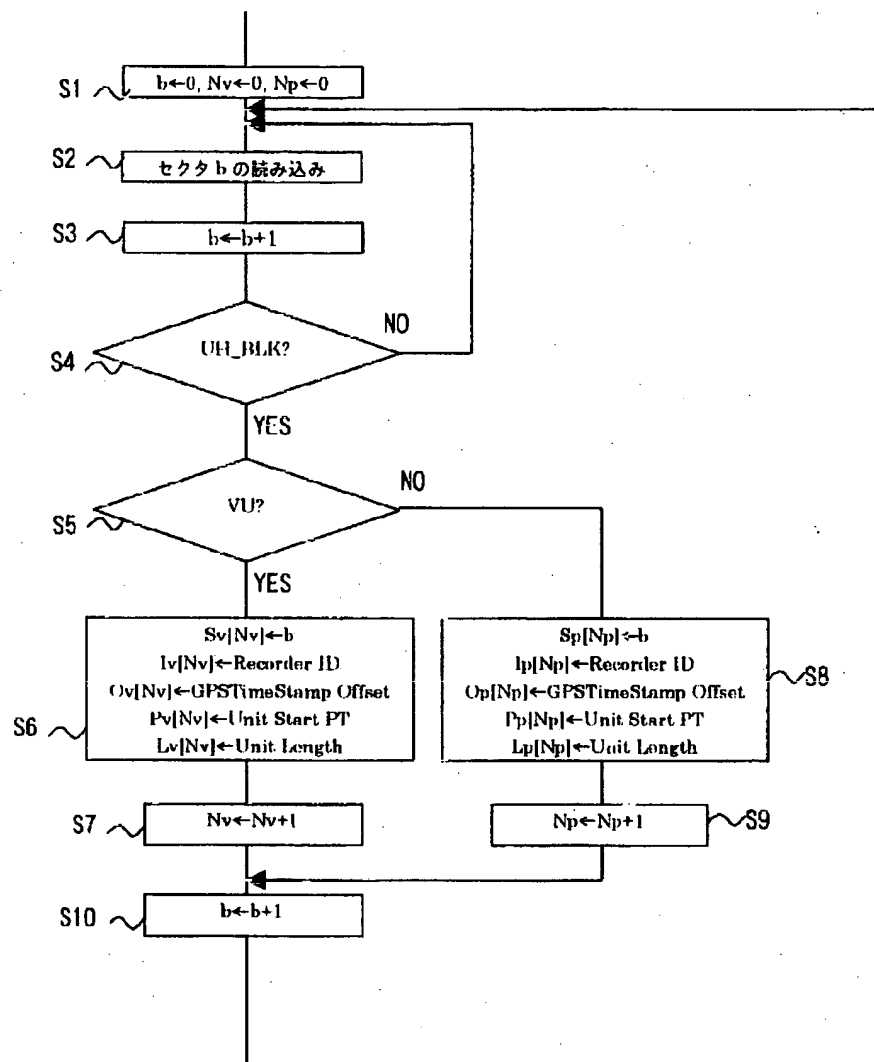
【図19】



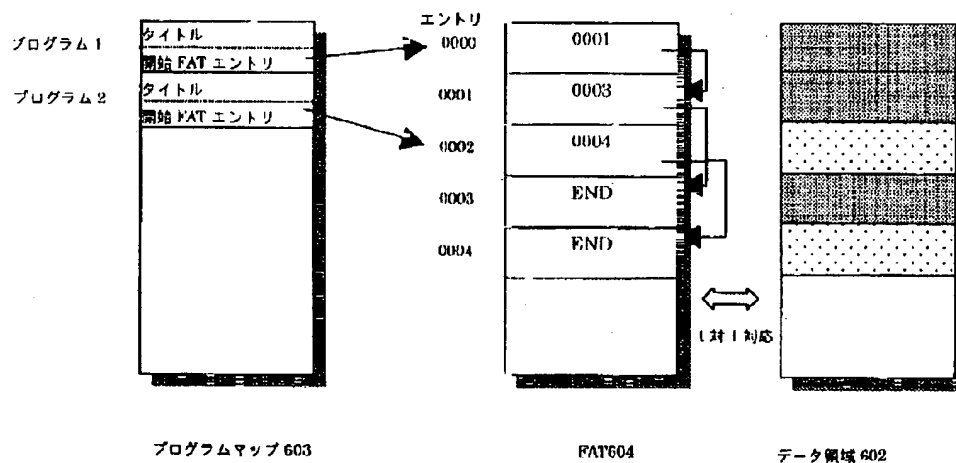
【図25】



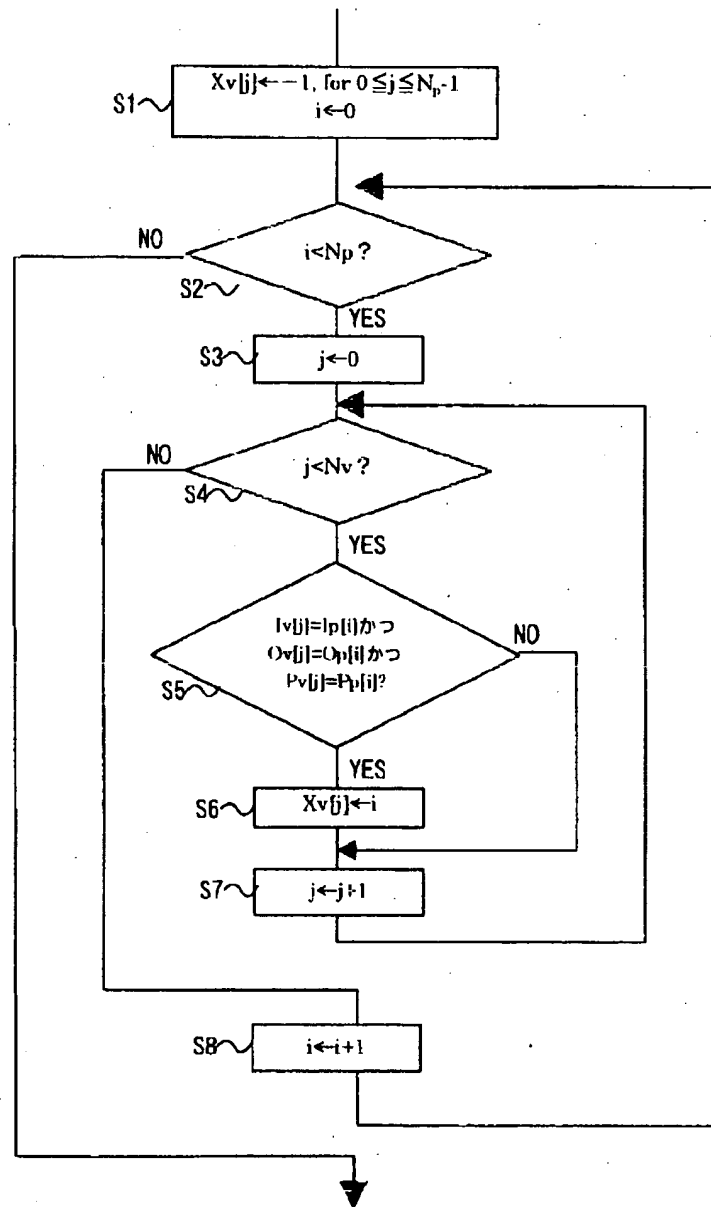
【図20】



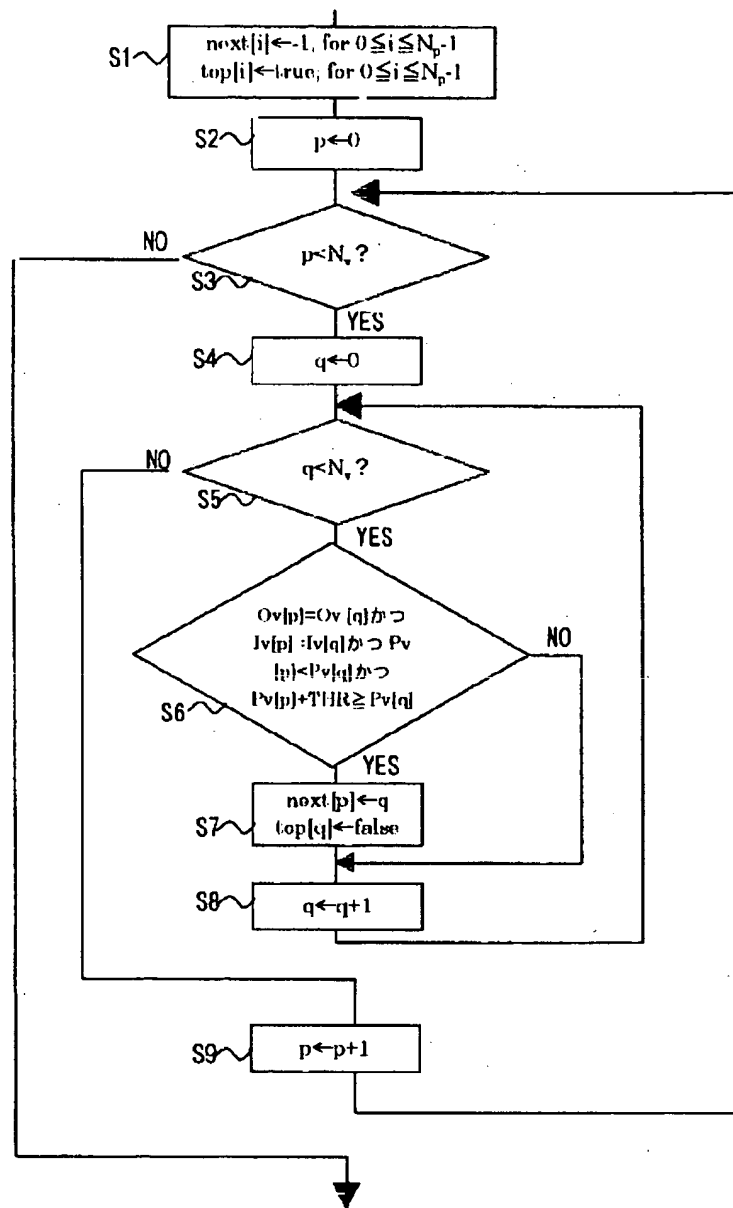
【図26】



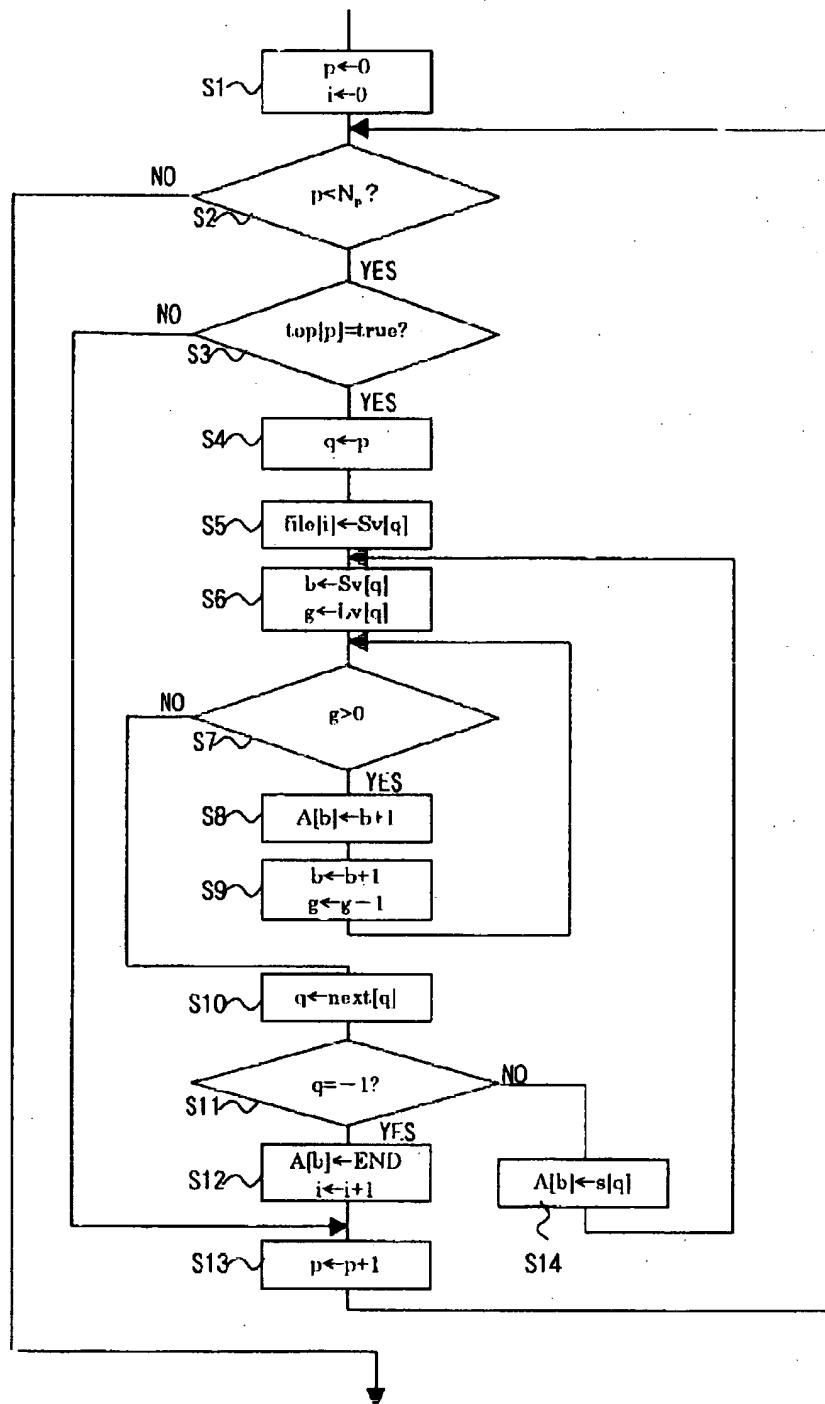
【図21】



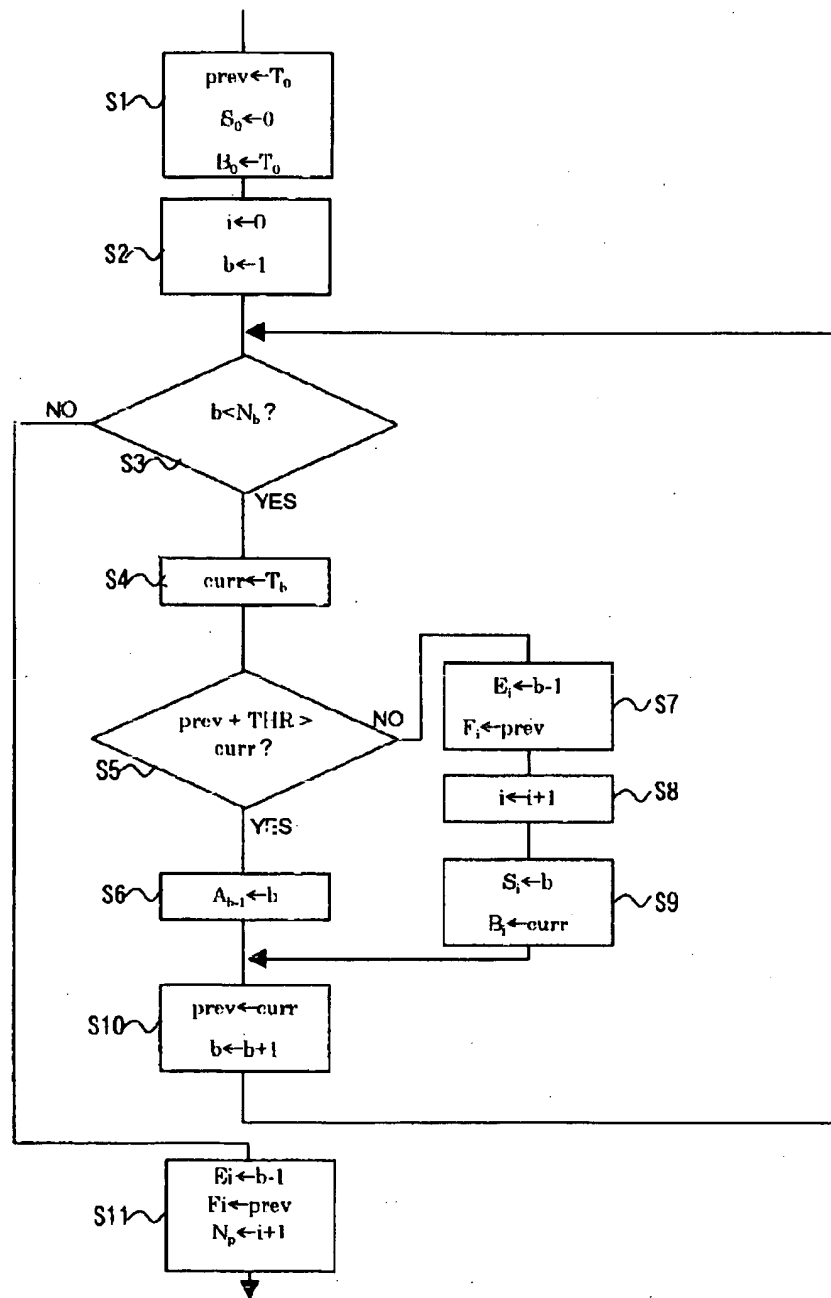
【図22】



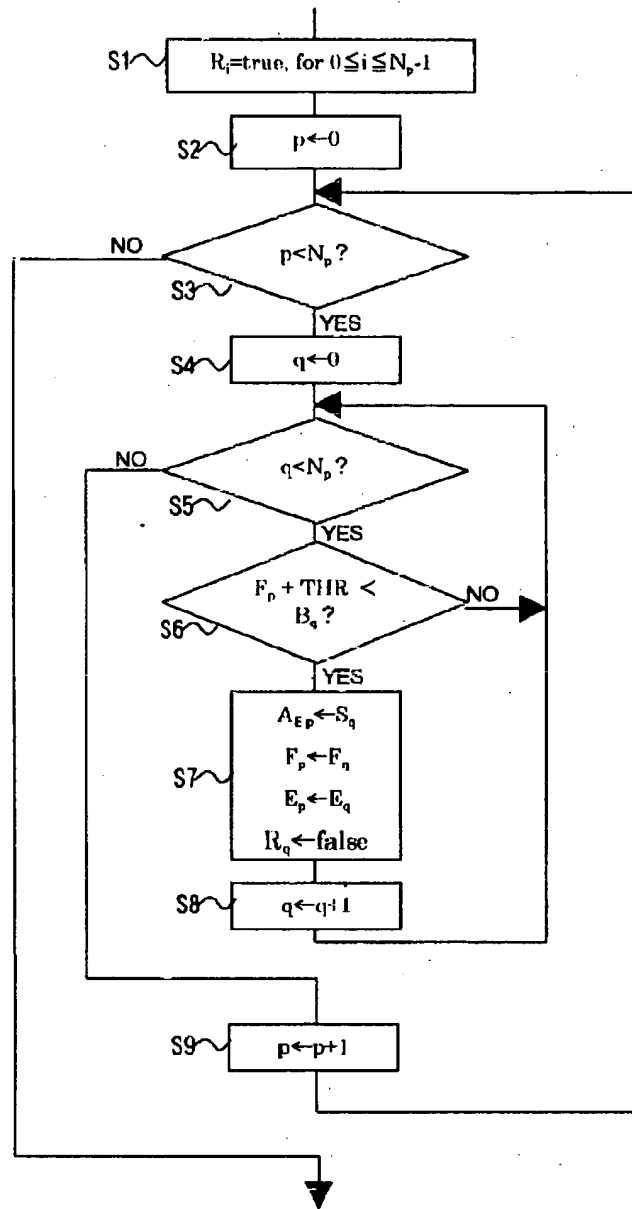
【図23】



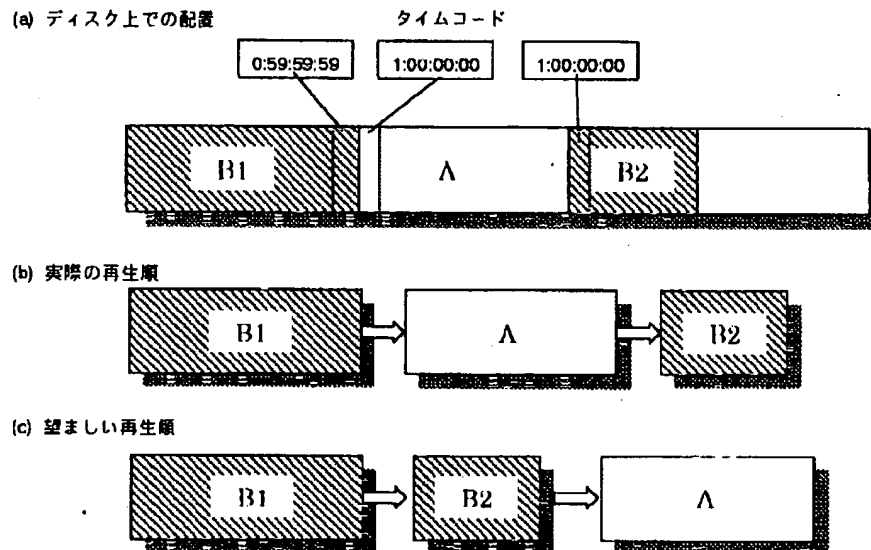
【図27】



【図28】



【図29】



フロントページの続き

(51)Int.Cl.⁷H04N 5/765
5/781

識別記号

FI

H04N 5/781
G11B 27/00
27/10510F
D
A

(参考)

(72)発明者 山口 孝好

大阪府大阪市阿倍野区長池町22番22号 シ
ャープ株式会社内

Fターム(参考) 5C053 FA20 FA23 GA06 GB06 GB07
GB37 HA33 JA03 JA24 KA04
KA24
5D044 AB07 BC06 CC04 DE39 DE45
DE53 DE57 EF05 FG18 GK12
5D077 AA30 BA15 CA02 CB06 DC16
DC22 DC37 DC39 EA34 HC12
HC26
5D110 AA17 AA29 DA03 DA12 DA15
DB03 DB17 DC05 DC06 DE06